|  |  |
| --- | --- |
| Program No | 12 |
| Roll No | 1333 |
| Unit | 3 |
| Program | Array Based Stack |

**Source Code:**

#include<iostream> #include<conio.h> #define MAX 5

using namespace std;

/\* Node Template - Not Required \*/

/\* Stack Template \*/ class AStack

{

int a[MAX]; int tos;

public:

AStack()

{

tos = -1;

}

void Push(int x); void Pop();

void Peek(); void Display(); int Full();

int Empty();

};

/\* Functions \*/

//Push

void AStack :: Push(int x)

{

if(Full())

{

}

else

{

}

}

cout << "Stack Overflow!" << endl;

tos++; a[tos] = x;

int AStack :: Full()

{

if(tos == MAX-1)

{

}

else

{

}

}

return 1;

return 0;

//pop

void AStack :: Pop()

{

if(Empty())

{

}

else

{

}

}

cout << "Stack Underflow!" << endl;

int t = a[tos]; tos--;

cout << "Element popped is: " << t << endl;

int AStack :: Empty()

{

if(tos == -1)

{

}

else

{

}

}

return 1;

return 0;

//Peek

void AStack :: Peek()

{

if(Empty())

{

}

else

{

}

}

cout << "Stack Underflow!" << endl;

cout << "Element at tos is: " << a[tos] << endl;

//Display

void AStack :: Display()

{

if(Empty())

{

}

else

{

cout << "Stack Underflow!" << endl;

cout << "Stack Contains: \n"; for(int i=tos; i>=0; i--)

{

cout << a[i] << endl;

}

}

}

/\* Menu \*/ int main()

{

int ch, num; AStack s;

while(1)

{

system("cls");

cout << "\*\*\*Array Based Stack\*\*\*\n\n";

cout << "1. Push an element" << endl; cout << "2. Pop an element" << endl; cout << "3. Peek at the stack" << endl; cout << "4. Display the stack" << endl; cout << "5. Exit" << endl;

cout << "\nEnter your choice: "; cin >> ch;

switch(ch)

{

case 1:

cout << "Enter a value: "; cin >> num; s.Push(num);

getch(); break;

case 2:

s.Pop();

getch(); break;

case 3:

s.Peek();

getch(); break;

case 4:

s.Display();

getch(); break;

case 5:

exit(1);

default:

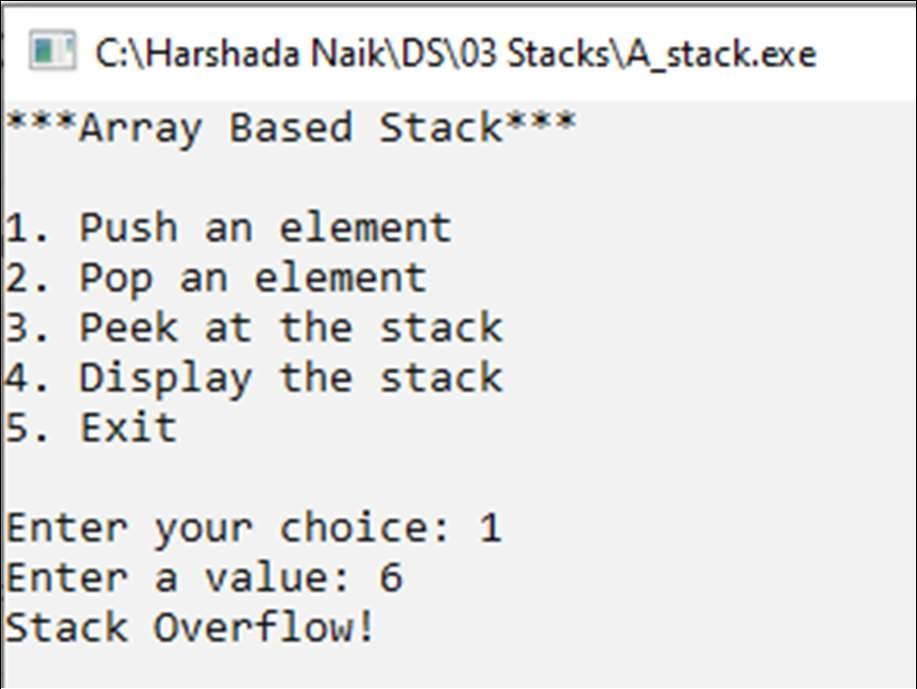
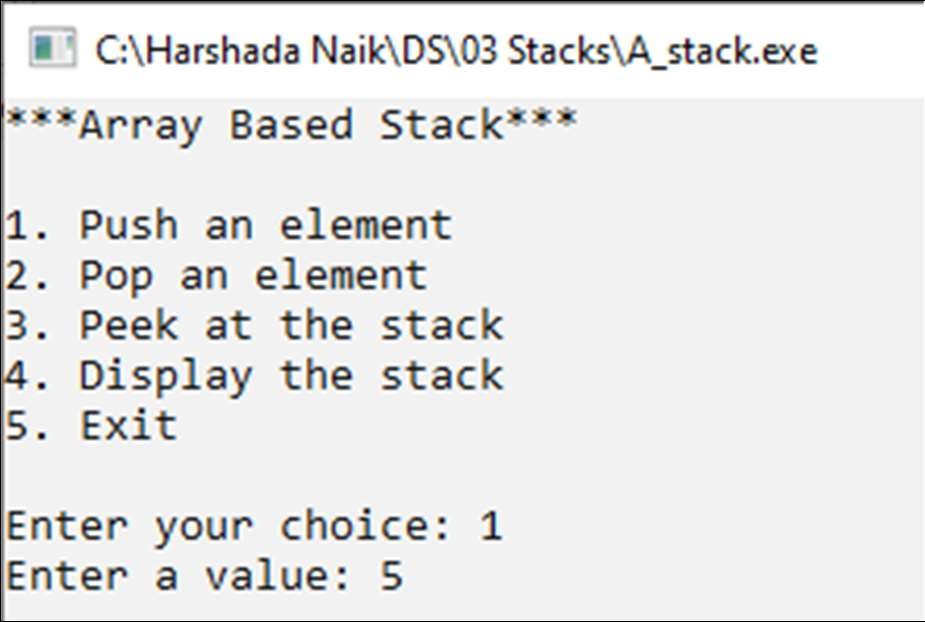
cout << "Incorrect Option"; getch();

} //end of switch

}// end of while

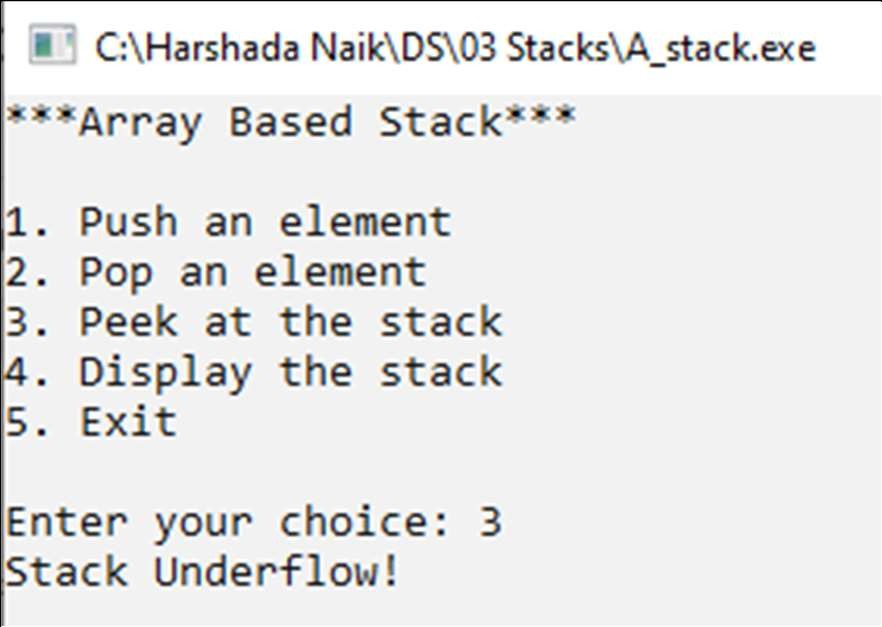
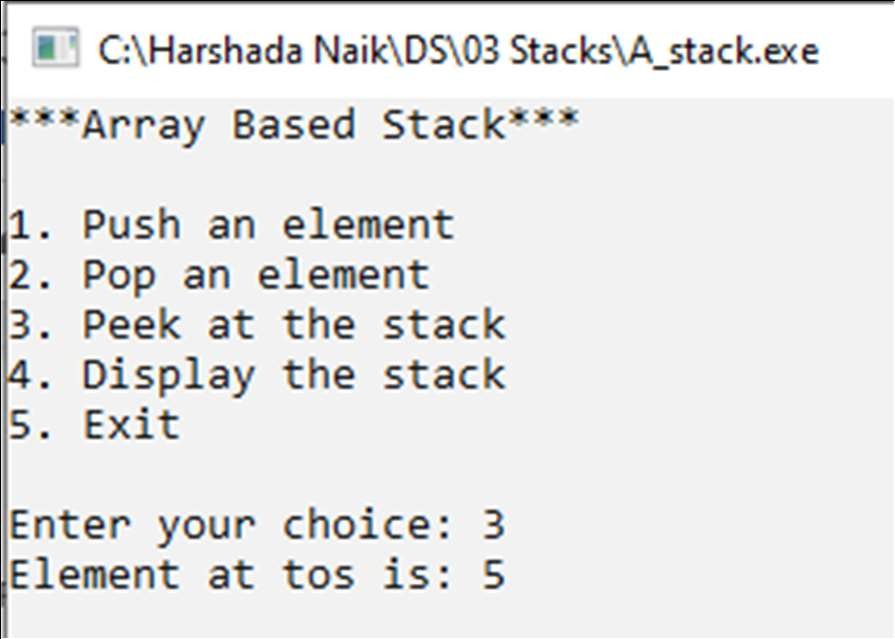
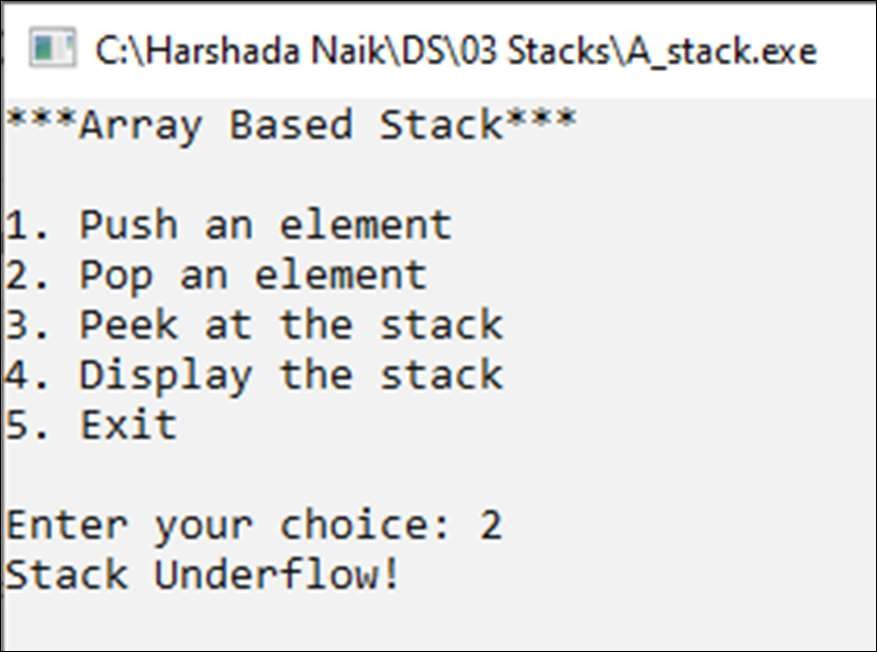
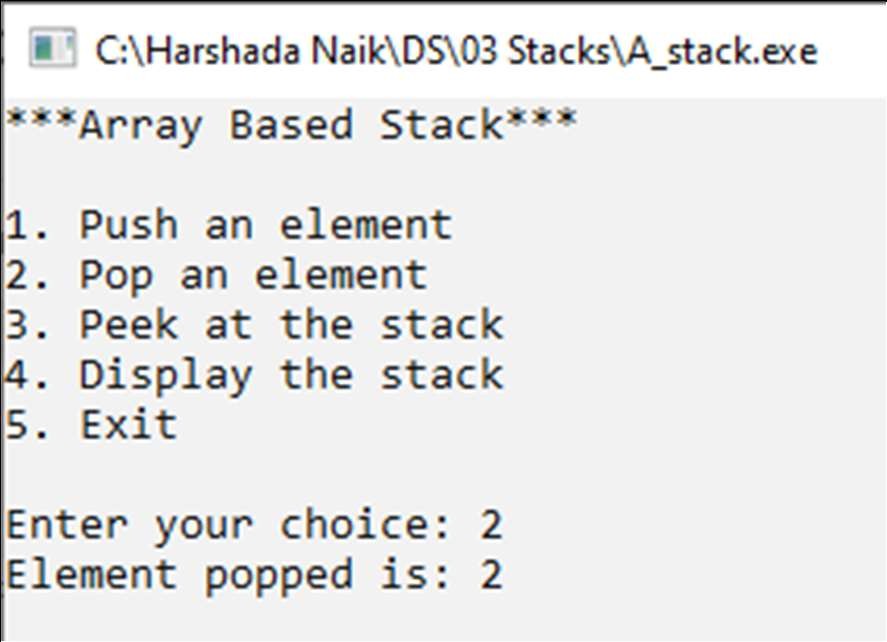
}//end of main

**Output:**



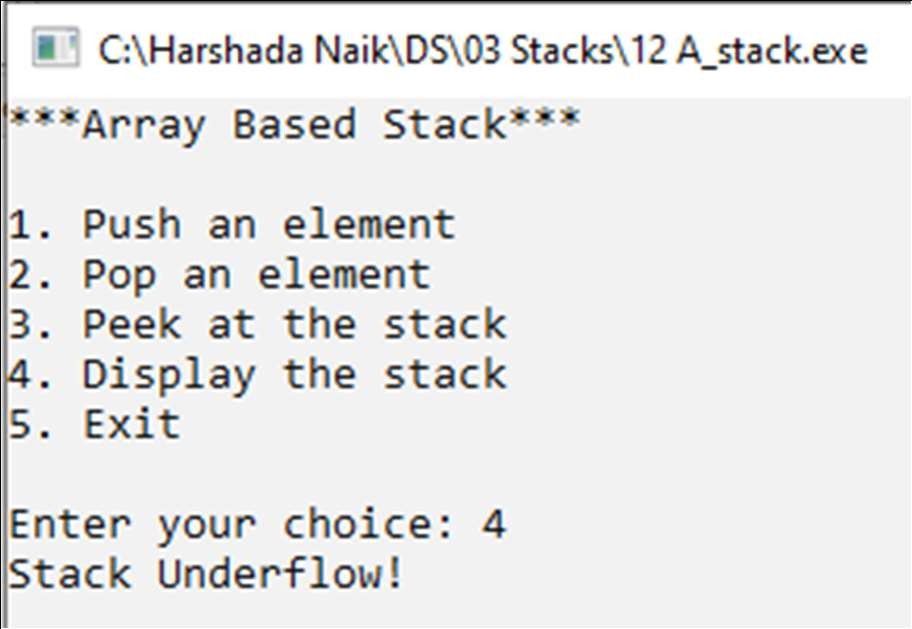
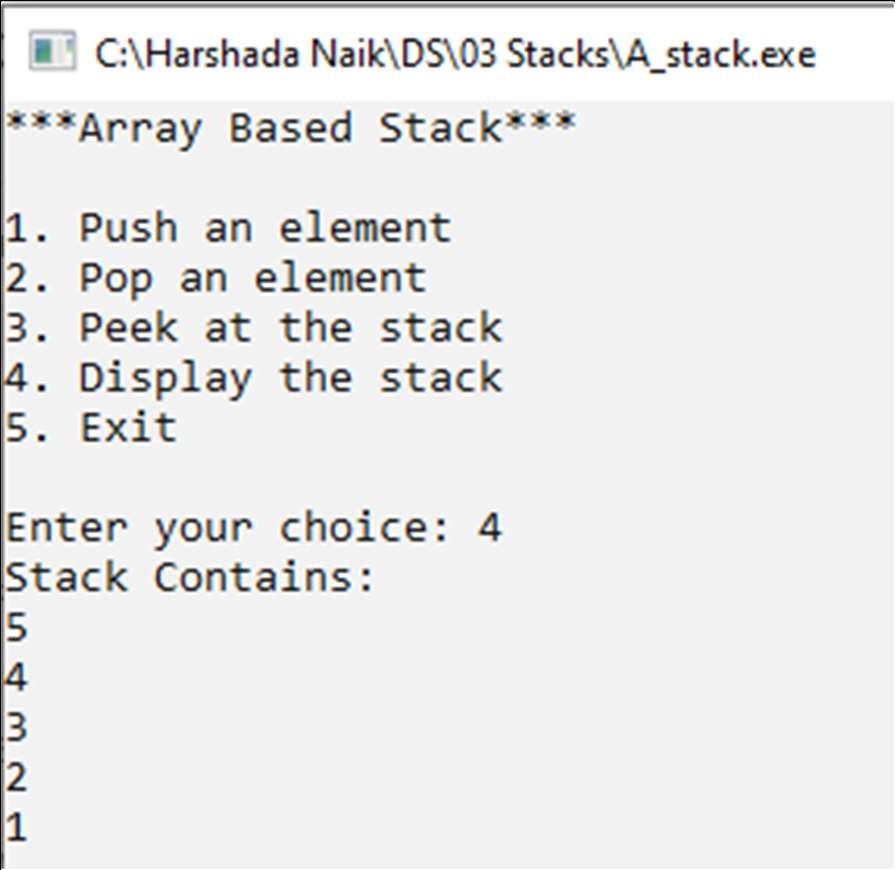
# MUMBAI EDUCATIONAL TRUST

**MET Institute of Computer Science**



# MUMBAI EDUCATIONAL TRUST

## MET Institute of Computer Science



|  |  |
| --- | --- |
| Program No | 13 |
| Roll No | 1333 |
| Unit | 3 |
| Program | Link Based Stack |

### Source Code:

#include<iostream> #include<conio.h>

using namespace std;

/\* Node Template \*/ class SNode

{

public:

int data; SNode \*next;

};

/\* Stack Template \*/ class LStack

{

SNode \*tos;

public:

LStack()

{

tos = NULL;

}

void Push(int x); void Pop();

void Peek(); void Display();

};

/\* Functions \*/

//Push

void LStack :: Push(int x)

{

//Create a node

SNode \*t = new SNode(); t->data = x;

t->next = NULL;

//First Node if(tos == NULL)

{

tos = t;

}

else // Insert at the head

{

t->next = tos; tos = t;

}

}

//Pop

void LStack :: Pop()

{

if(tos == NULL)

{

}

else

{

cout << "Stack Underflow!";

SNode \*t = tos; tos = tos->next;

cout << "Element popped is: " << t->data; delete t;

}

}

//Peek

void LStack :: Peek()

{

if(tos == NULL)

{

}

else

{

}

}

cout << "Stack Underflow!";

cout << "Element at tos is: " << tos->data;

//Display

void LStack :: Display()

{

if(tos == NULL)

{

}

else

{

cout << "Stack Underflow!";

cout << "Stack contains: \n";

SNode \*tmp = tos; while(tmp != NULL)

{

cout << tmp->data << " -> " << endl; tmp = tmp->next;

}

cout << "NULL";

}

}

/\* Menu \*/ int main()

{

int ch, num; LStack L;

while(1)

{

system("cls");

cout << "\*\*\*List Based Stack\*\*\*\n" << endl;

cout << "1. Push an element" << endl; cout << "2. Pop an element" << endl; cout << "3. Peek at the stack" << endl; cout << "4. Display the stack" << endl; cout << "5. Exit" << endl;

cout << "\nEnter your choice: "; cin >> ch;

switch(ch)

{

case 1:

cout << "Enter the value: "; cin >> num;

L.Push(num);

getch(); break;

case 2:

L.Pop();

getch(); break;

case 3:

L.Peek();

getch(); break;

case 4:

L.Display();

getch(); break;

case 5:

exit(1);

default:

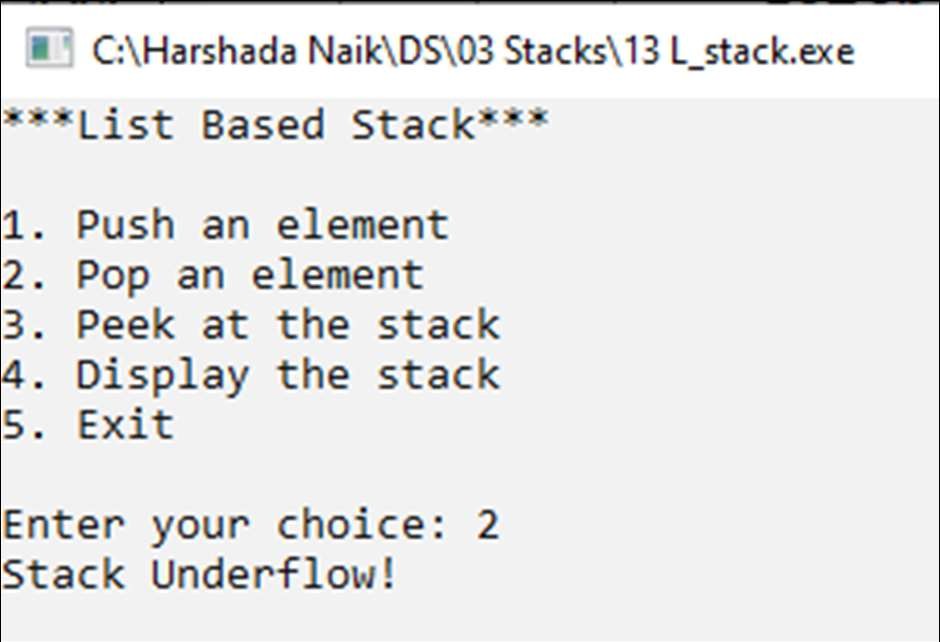
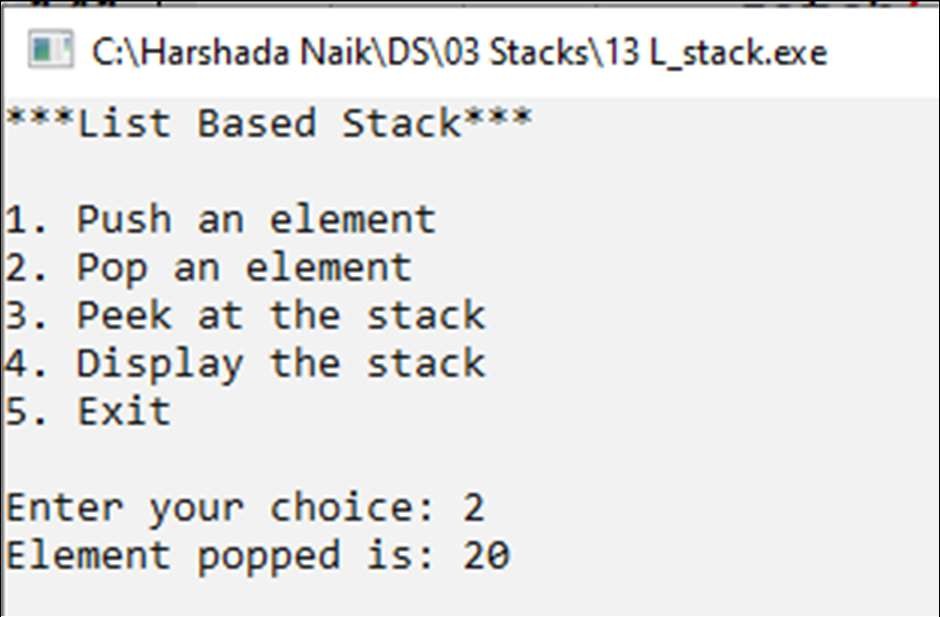
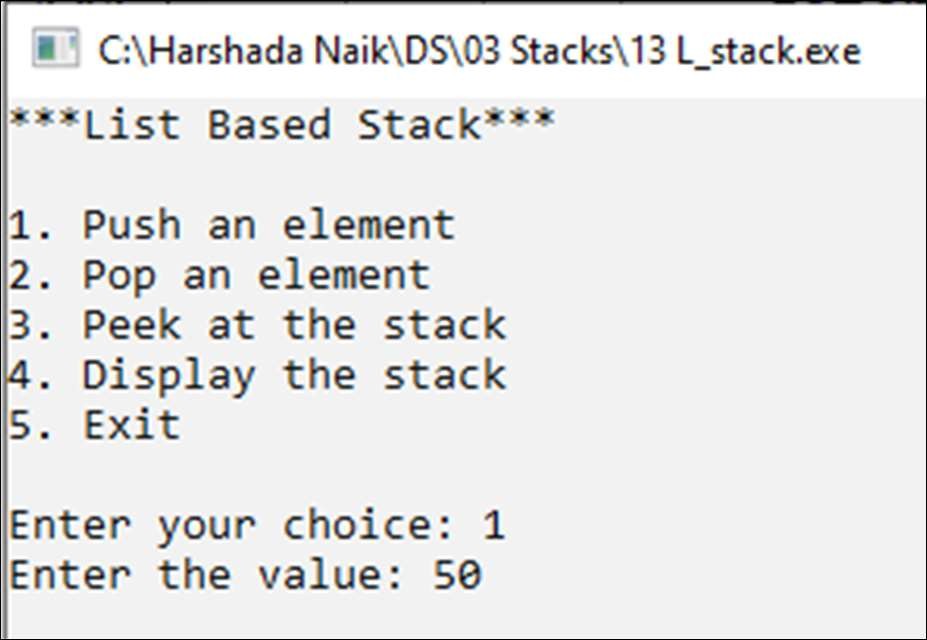
cout << "Incorrect option"; getch();

}//end of switch

}//end of while

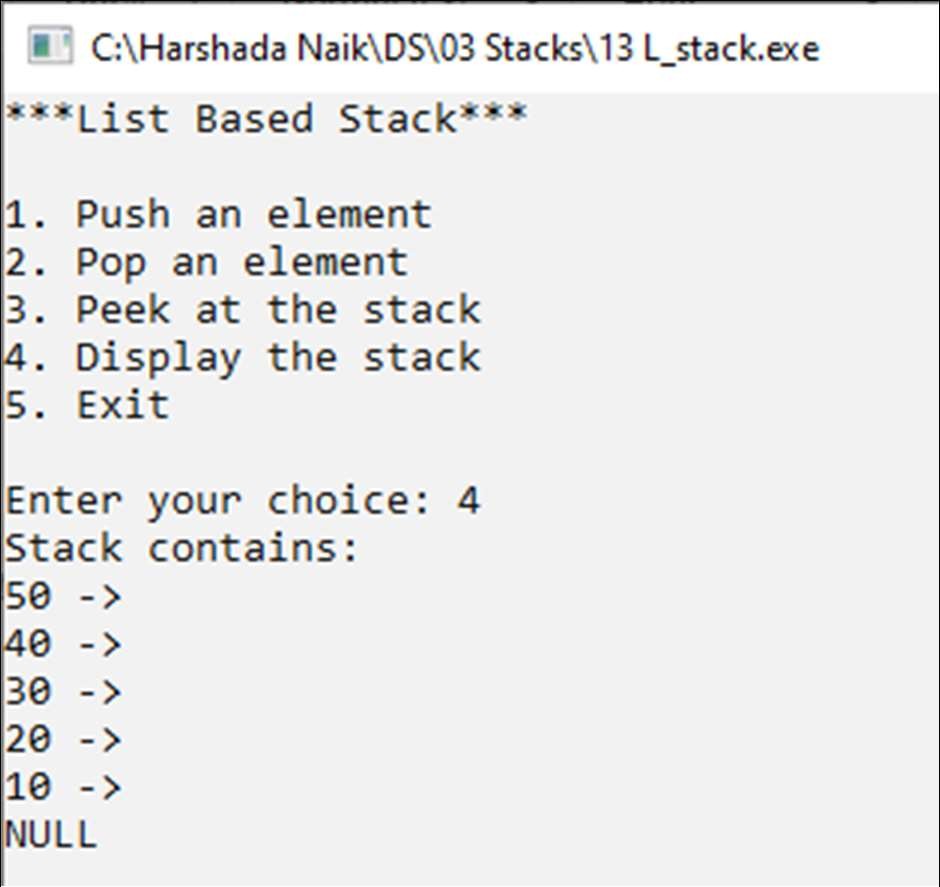
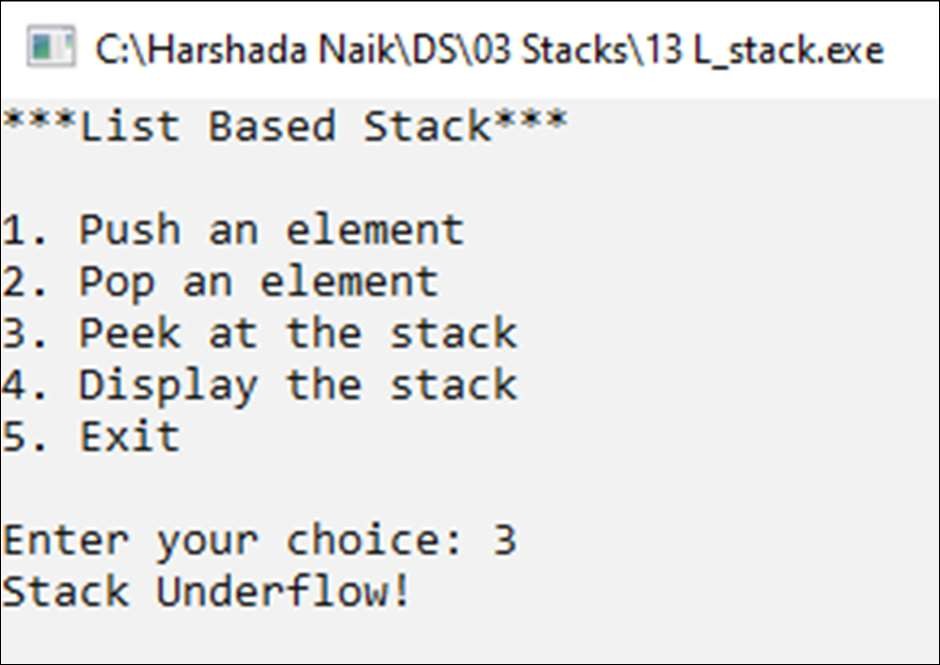
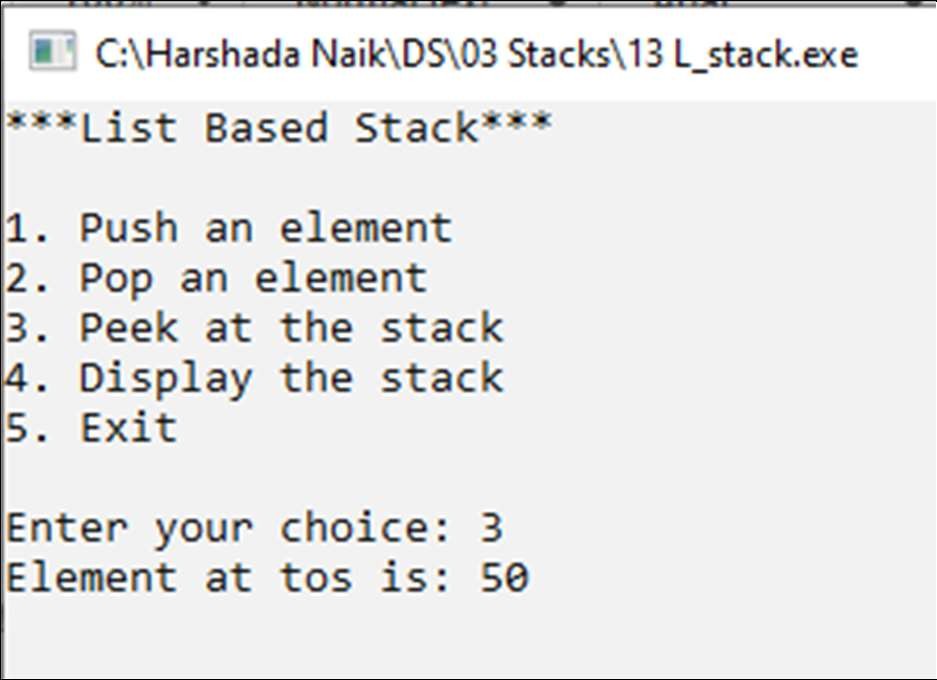
}//end of main

**Output:**



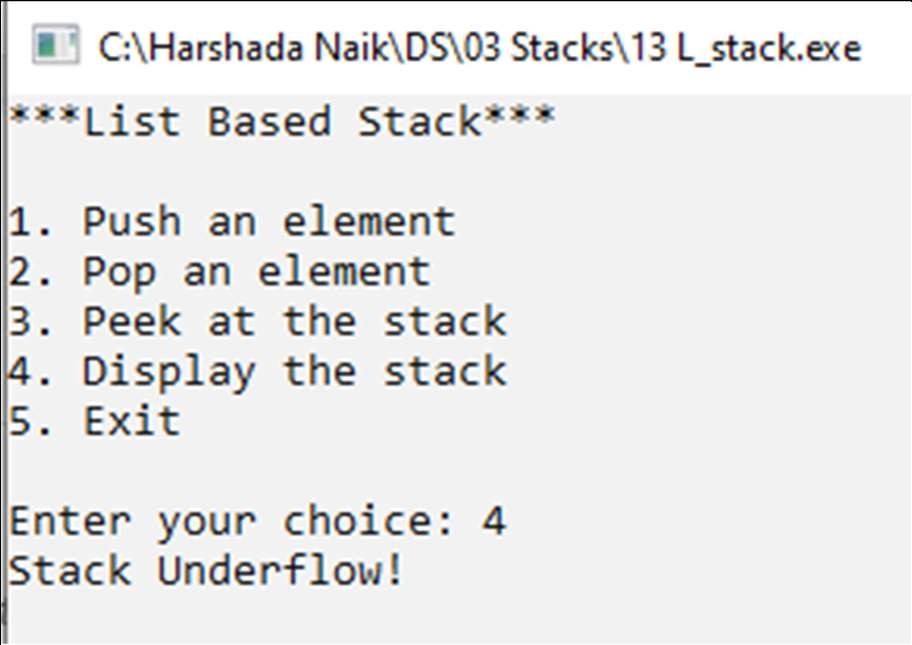
# MUMBAI EDUCATIONAL TRUST

**MET Institute of Computer Science**



# MUMBAI EDUCATIONAL TRUST

## MET Institute of Computer Science



|  |  |
| --- | --- |
| Program No | 14 |
| Roll No | 1333 |
| Unit | 3 |
| Program | Balancing of Parenthesis |

### Source Code:

#include<iostream> #define SIZE 20

using namespace std;

/\* 2. Stack Template \*/ class Stack

{

char a[SIZE]; int tos;

public:

Stack()

{

}

tos = -1;

void Push(char x); char Pop();

int Full(); int Empty();

};

/\* functions \*/

//push

void Stack :: Push(char x)

{

if(Full())

{

}

else

{

}

}

cout << "Stack Overflow!";

tos++; a[tos] = x;

//pop

char Stack :: Pop()

{

if(Empty())

{

cout << "Stack Underflow!"; return -1;

}

else

{

}

}

char t = a[tos]; tos--;

return t;

//Full

int Stack :: Full()

{

return (tos == SIZE-1?1:0);

}

//Empty

int Stack :: Empty()

{

return (tos == -1?1:0);

}

/\* Main Application \*/ int main()

{

char expr[20], temp; int i, flag = 0;

Stack s;

system("cls");

cout << "program to check if the parenthesis are balanced in an expression\n\n";

cout << "Enter an expression: "; cin >> expr;

while(expr[i] != '\0')

{

if(expr[i] == '(' || expr[i] == '{' || expr[i] == '[')

{

s.Push(expr[i]);

}//end of if open bracket

if(expr[i] == ')' || expr[i] == '}' || expr[i] == ']')

{

if(s.Empty()) //Takes care of extra bracket at the end

{

flag = 1; break;

}

temp = s.Pop();

if((temp == '(' && expr[i] == ')') ||

(temp == '{' && expr[i] == '}') ||

(temp == '[' && expr[i] == ']'))

{

i++;

continue;

}

else //Brackets don't match

{

flag = 1; break;

}

}//end of if close bracket

i++;

}//end of while

if(s.Empty() && flag == 0)

{

}

else

{

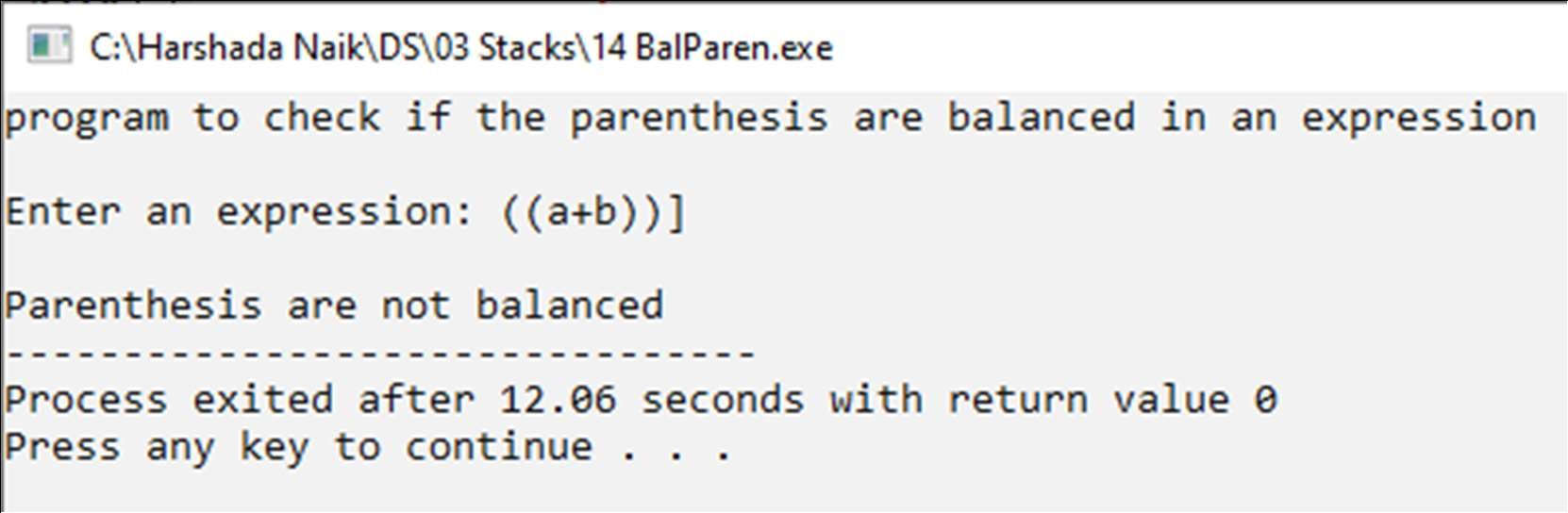
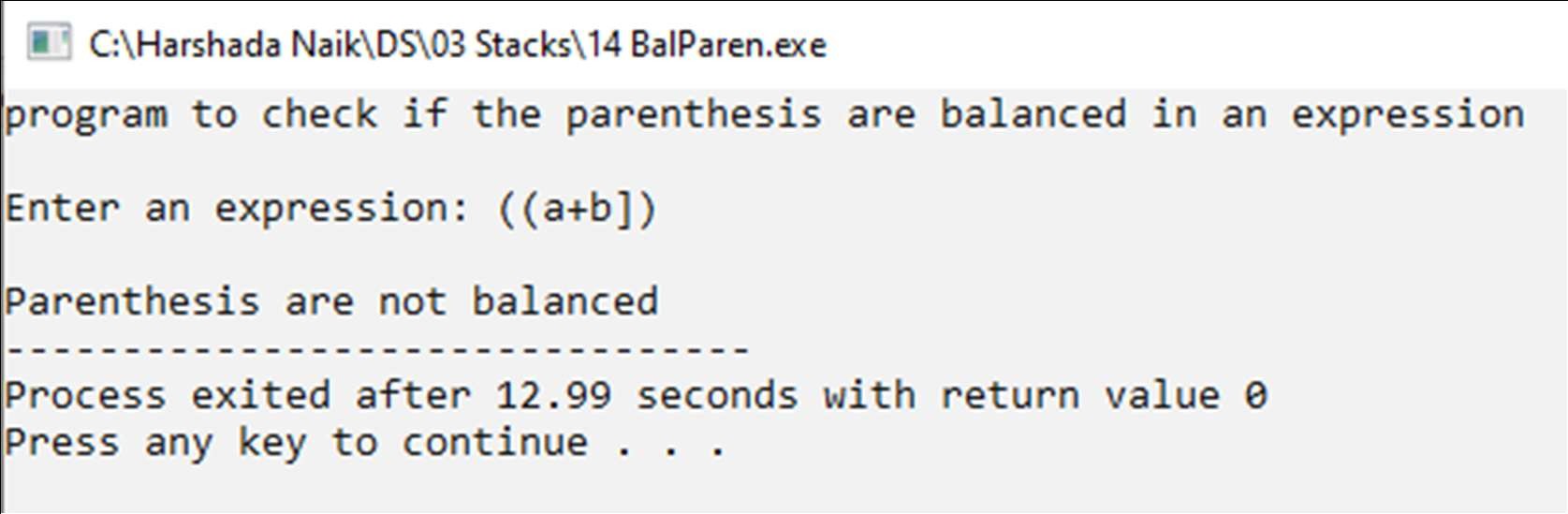
}

}//end of main

### Output:

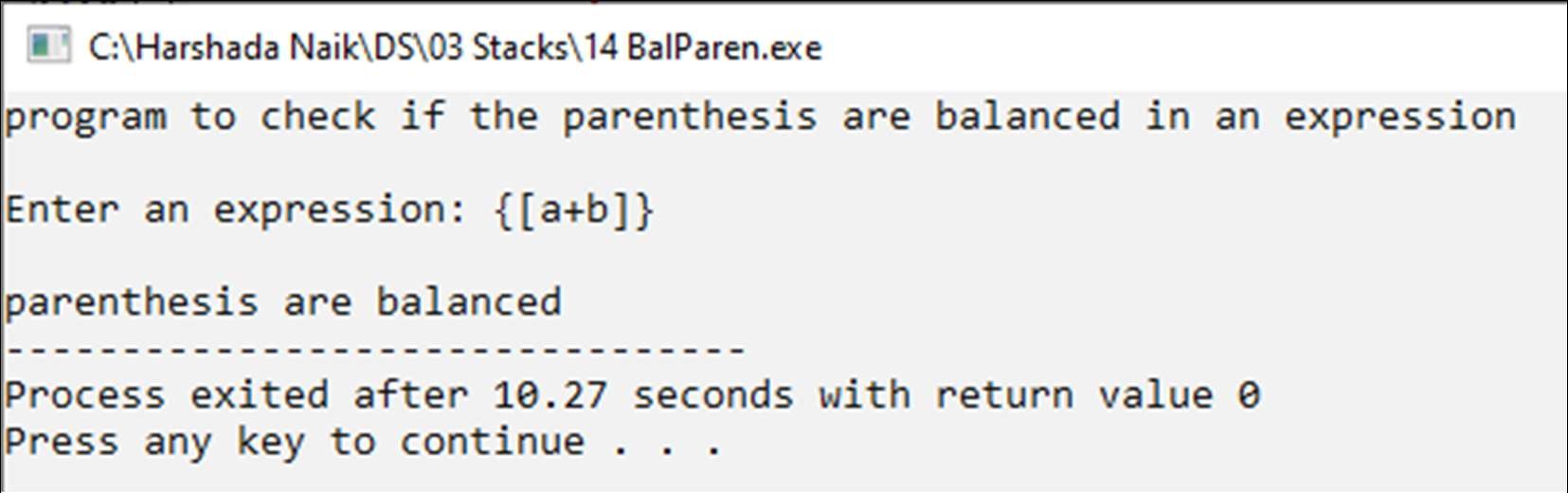
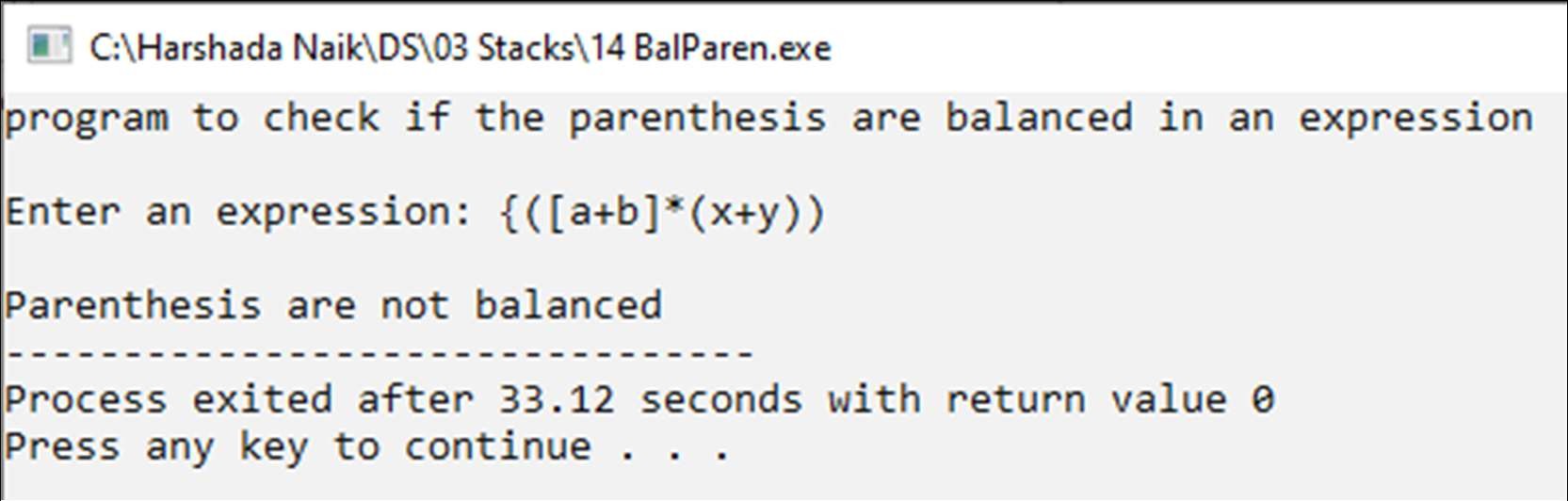
cout << "\nparenthesis are balanced";

cout << "\nParenthesis are not balanced";



# MUMBAI EDUCATIONAL TRUST

## MET Institute of Computer Science



|  |  |
| --- | --- |
| Program No | 15 |
| Roll No | 1333 |
| Unit | 3 |
| Program | Evaluation of Postfix Expression |

### Source Code:

#include"F\_Stack.cpp" int main()

{

char postfix[20], temp; int i = 0;

float a, b, result; Stack s;

cout << "\*\*\* Evaluation of postfix expression \*\*\*\n\n"; cout << "Enter a postfix expression: ";

gets(postfix);

while(postfix[i] != '\0')

{

temp = postfix[i];

if(temp == ' ') //for blank space

{

i++;

continue;

cout << "In Space" << endl;

}

if(isdigit(temp))

{

float x;

x = temp - 48.0; s.Push(x);

}

else //operator

{

b = s.Pop();

a = s.Pop();

switch(temp)

{

case '+':

result = a+b; break;

case '-':

result = a-b; break;

case '\*':

result = a\*b; break;

case '/':

result = a/b; break;

}//end of switch s.Push(result);

}//end of else i++;

}//end of while result = s.Pop();

cout << "Result of the evaluation is: " << result;

}//end of main

**FStack.cpp**

#include<iostream> #include<ctype.h> #include<conio.h> #define SIZE 20

using namespace std;

/\* Stack template \*/ class Stack

{

float a[SIZE]; int tos;

public:

Stack()

{

}

tos = -1;

void Push(float x); float Pop();

int Full(); int Empty();

};

/\* Functions \*/

//push

void Stack :: Push(float x)

{

if(Full())

{

cout << "Stack Overflow!";

}

else

{

}

}

tos++; a[tos] = x;

float Stack :: Pop()

{

if(Empty())

{

}

else

{

}

}

cout << "Stack Underflow!"; return -1;

float temp = a[tos]; tos--;

return temp;

//full

int Stack :: Full()

{

return (tos == SIZE-1?1:0);

}

//empty

int Stack :: Empty()

{

return (tos == -1?1:0);

}

### Output:

